

# Paiza.ioを使った Pythonプログラミング

# Paiza.ioへのログイン

<https://paiza.io> へアクセスして、

ログインをクリック



# Googleアカウントでログインをクリック

ログイン



メールアドレス

Required

パスワード

Required

☐ 保存する



パスワード忘れはこちら  
確認メールの再送信

※スマホなどで本人認証が行われる場合があります。

※アクセスが拒否された場合は、いったんブラウザを全部閉じてから再試行してみてください

# 身長データを読み込む プログラムの作成

## 1)新規コードをクリック



## 2)Phyton3編集モードに切り替える

「PHP」と書かれた  
左上のボタン  
をクリック



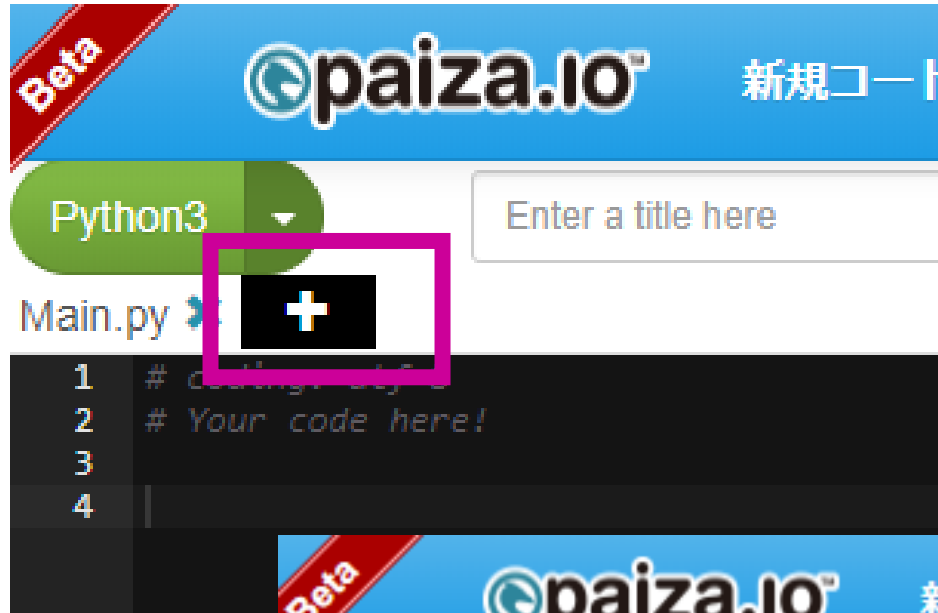
表示されたウィンドウでPhyton3を選択

# ボタンがPython3に切り替わったことを確認

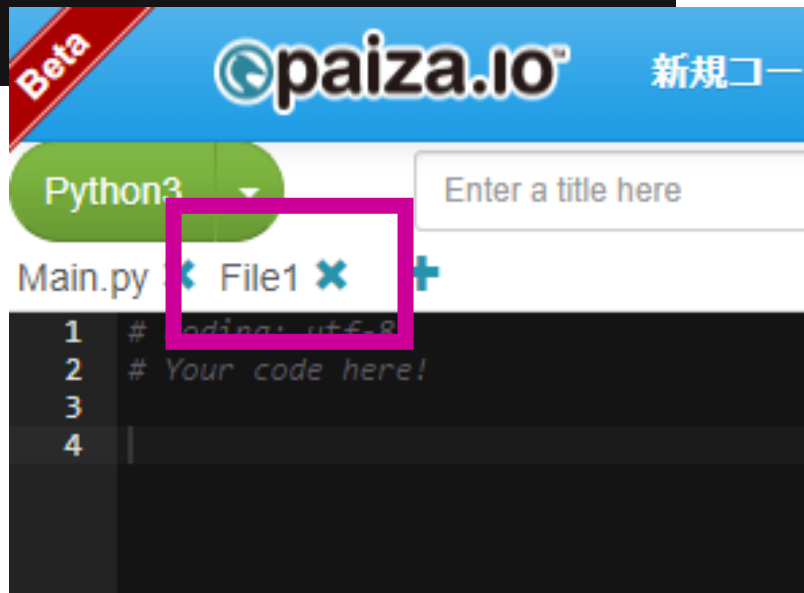


たまたま、ちゃんとPython3モードになっているか  
確認してください

### 3) データ入力用のファイルをpaiza.io上につくる

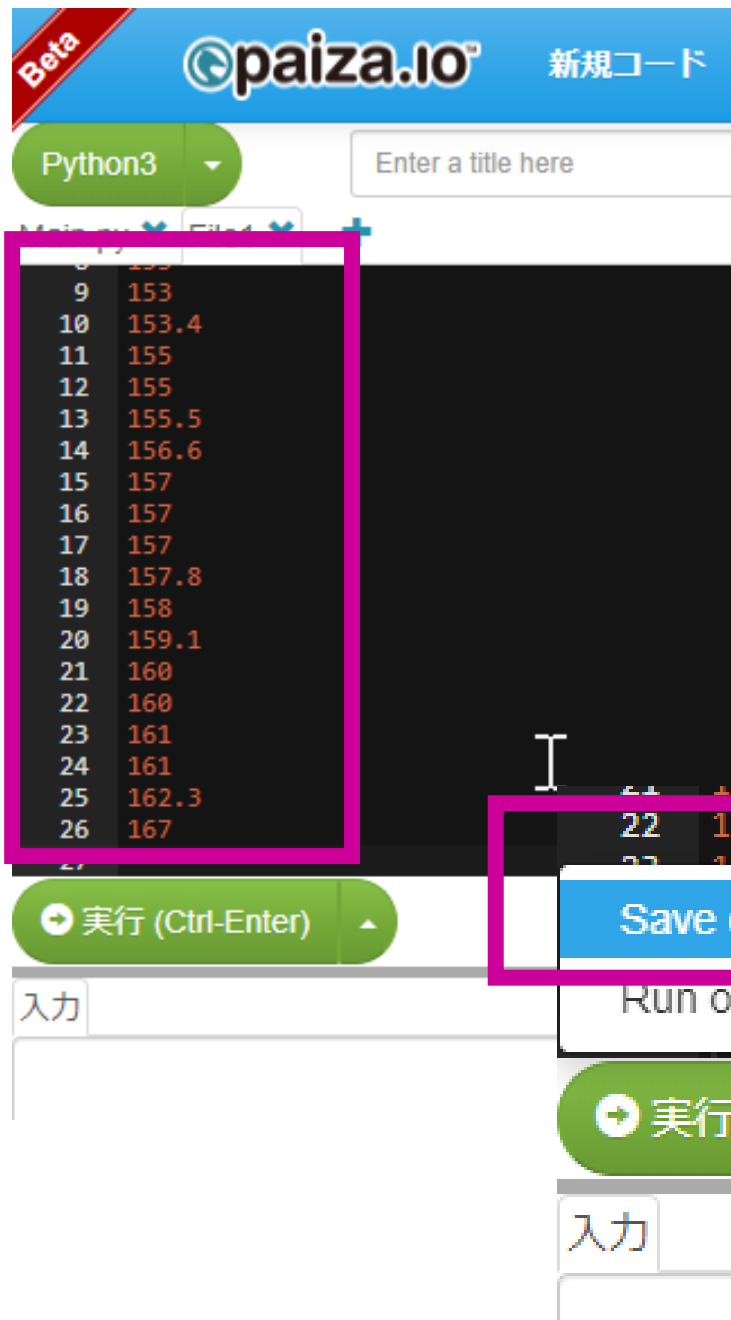


ここの「+」を押すと、



新たに「File1」という  
タブができる。

名前の変更も可能ですが、  
ここではこの名前のまま  
使います。



File1のタブをクリックします。

File1のウィンドウ内に、身長データ  
(数字)をコピー&ペーストします

「実行ボタン」の隣の三角形を  
クリックして、「Save only」を  
選択

これでデータが  
paiza.ioに保  
存されました



# Main.pyタブをクリックします



# Main.pyを以下のように編集します

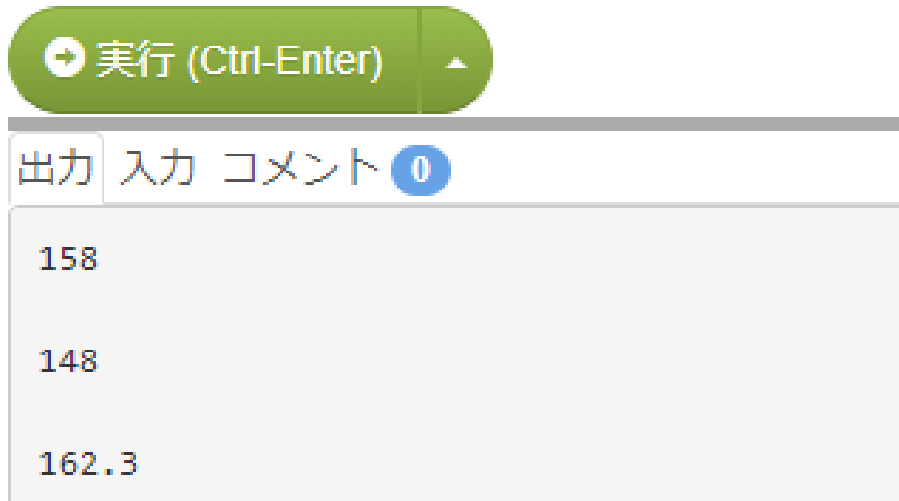
```
1  # coding: utf-8
2  # Your code here!
3
4  data = open("File1", "r")
5
6  for line in data:
7      print(line)
8
9  data.close()
10
```

openでファイルを開いて、dataで取り扱えるようにし、for～inで内容を一行ずつlineに読み込み、それをprintで画面上に表示するプログラムです。最後に使ったdataを閉じて終わり。

openの”r”は、読み込み(read)モードで開くことを意味しています。

forで繰り返される内容(7行目のprintの行)は、字下げ(インデント)して書きます。pythonは、この字下げに大きな意味があり、**字下げしないとエラー**になります。

「実行」ボタンを押すと、結果が出力画面に表示されます。



※データによって図の数値とは異なる場合があります。

## コア機能と補助機能

## 補助機能2 コメント

## コア機能2 変数

## コア機能1 入出力

(画面へ出力)  
(コア機能3  
命令でもある)

コア機能1  
入出力  
(ファイルから入力)  
(補助機能3  
関数でもある)

コア機能5  
制御構造  
(繰り返し)

## コア機能3 命令

```
1 # coding: utf-8
2 # Your code here!
3
4 data = open("File1", "r")
5
6 for line in data:
7     print(line)
8
9 data.close()
10
```

# 何かおかしいなところがあると、下記のようにエラーが出ます

Python3

Main.py ☒ File1 ☒ +

Runtime error(Exit status:1)

```
1 # coding: utf-8
2 # Your code here!
3
4 data = open("File1", "r")
5
6 for line in data:
7     print(line)
8
9 date.close()
10
```

→ 実行 (Ctrl-Enter) Pythonを学ぶ | プログラミング力診断

望 ⓘ ⚙ ⌚ ↺ ☆ 👥 📁 ↻

出力 実行時エラー 入力 コメント 0 (0.16 sec)

Traceback (most recent call last):  
File "Main.py", line 9, in <module>  
date.close()  
NameError: name 'date' is not defined

} 補助3 デバッグ

エラーの原因を探る大きなヒントになるので、迷わず活用しましょう。  
9行目にエラーがあり、「dateが定義されていない」と言われています。変数名の「data」と書くべきところ、スペルミスして「date」と書いていたようです。

# 読み込んだデータの合計をだす

Main.pyを改良して次のように編集します。

```
4 data = open("File1", "r")
5
6 total = 0
7
8 for line in data:
9     total = total + float( line )
10    print(line)
11
12 data.close()
13
14 print( total )
15
```

合計を覚えておくための変数totalを作り(初期値0)、一行読み込まれるたびに自分自身に足してゆきます。繰り返す内容(9行目と10行目)は、同じ字下げになっていることに注意！ floatは文字データとして読み込まれた身長データを数値データに変換する関数です。最後にtotalを表示して終了しています。

# コア機能と補助機能

コア機能2  
変数

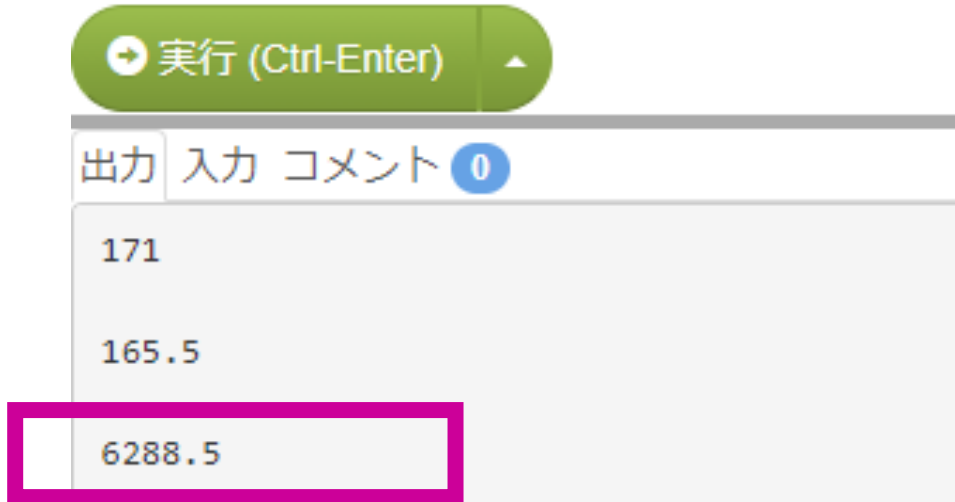
```
4 data = open("File1", "r")
5
6 total = 0
7
8 for line in data:
9     total = total + float( line )
10    print(line)
11
12 data.close()
13
14 print( total )
15
```

コア機能3  
演算子

補助機能1  
関数

(float関数は、あらかじめ用意されている関数)

# 「実行」ボタンを押して結果を確認



※データによって図の数値とは異なる場合があります。

## 合計が表示されました。



# 平均を出すために、データの個数を数える

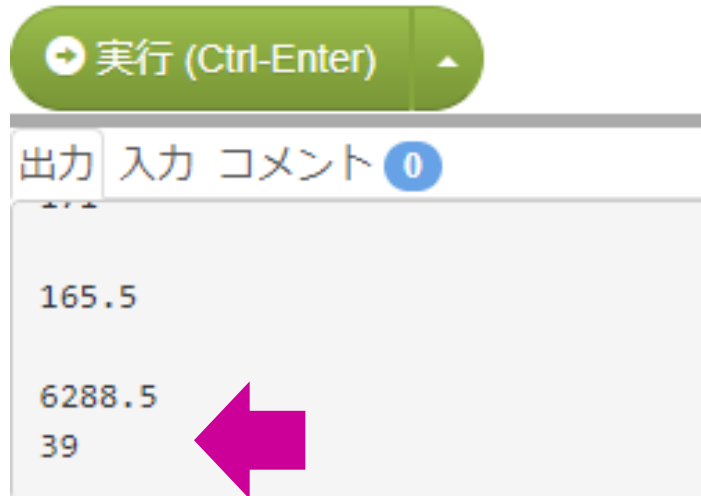
Main.pyを編集します。

```
4 data = open("File1", "r")
5
6 total = 0
7 num = 0
8
9 for line in data:
10     total = total + float( line )
11     num = num + 1
12     print(line)
13
14 data.close()
15
16 print( total )
17 print( num )
18
```

データの個数を覚えておくためのnum変数を作り、初期値0を設定しておきます。データが読み込まれるたびにnumを1ずつ増やします。

最後に、画面に表示しています。

# 「実行」ボタンを押して結果を確認



※データによって図の数値とは異なる場合があります。

## 個数が表示されました

Excelでの計算結果と一致しているか確認しましょう。

# 合計とデータの個数から 平均値を計算する

Main.pyをさらに改良します

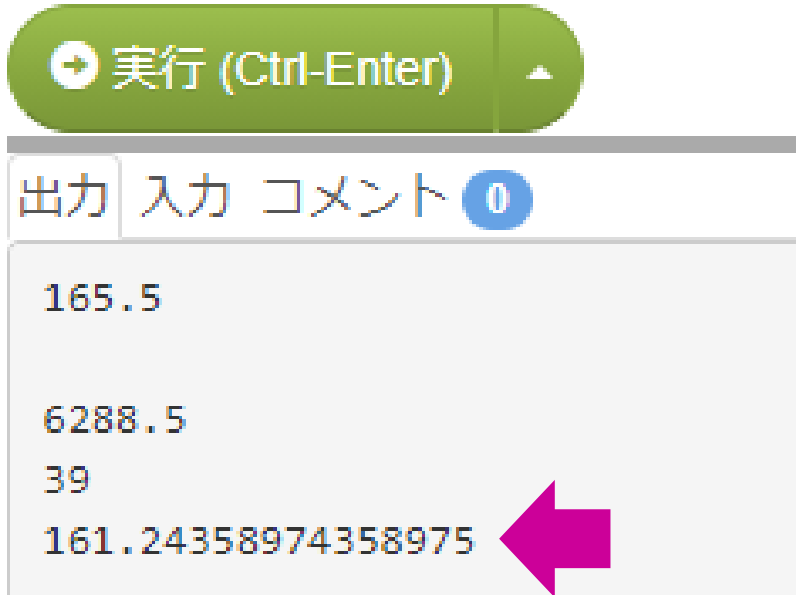
```
4 data = open("File1", "r")
5
6 total = 0
7 num = 0
8
9 for line in data:
10     total = total + float( line )
11     num = num + 1
12     print(line)
13
14 data.close()
15
16 print( total )
17 print( num )
18
19 ave = total / num
20
21 print( ave )
```

total÷numを計算して、それをave変数に保持します。

割り算(÷)は、「/」記号です。

最後にprintで画面に表示します。

# 「実行」ボタンを押して結果を確認



※データによって図の数値とは異なる場合があります。

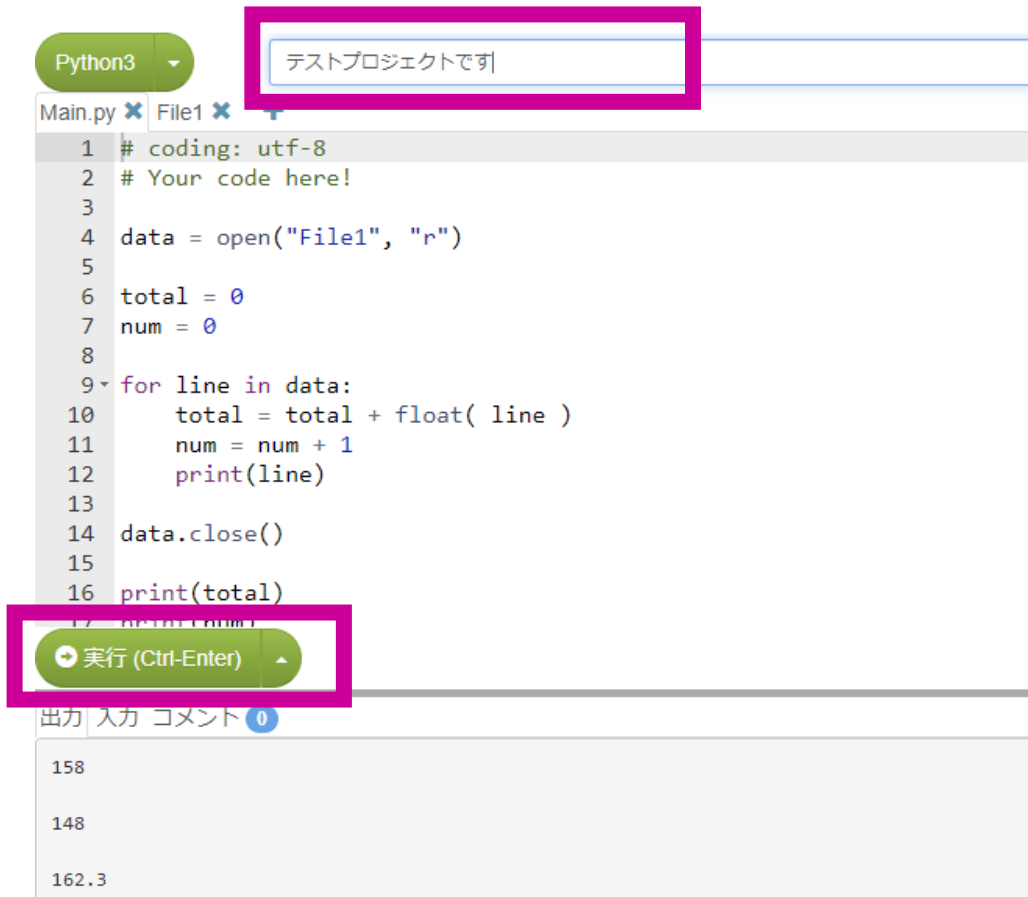
## 平均値が表示されました。

Excelでの計算結果と一致しているか確認しましょう。

# 完了！

**補足情報**

# プログラムの保存



The screenshot shows a Python IDE interface. At the top left, there is a dropdown menu set to 'Python3'. To its right is a text input field containing 'テストプロジェクトです' (This is a test project), which is highlighted with a pink rectangular box. Below the input field is a code editor with a file named 'Main.py'. The code in the editor is as follows:

```
1 # coding: utf-8
2 # Your code here!
3
4 data = open("File1", "r")
5
6 total = 0
7 num = 0
8
9 for line in data:
10     total = total + float( line )
11     num = num + 1
12     print(line)
13
14 data.close()
15
16 print(total)
```

At the bottom of the code editor, there is a green button with a play icon and the text '実行 (Ctrl-Enter)', which is also highlighted with a pink rectangular box. Below the code editor is an output console with tabs for '出力' (Output), '入力' (Input), and 'コメント' (Comments). The '出力' tab is selected, and it shows the following output:

```
158
148
162.3
```

プログラムには、適当なタイトル (日本語可)をつけることができます。

「実行」をするか、「Save Only」を行うと、プログラムは自動的にpaiza.ioに保存されます。

# 保存したプロジェクトの呼び出し

The screenshot shows the paiza.io website interface. The top navigation bar includes a 'Beta' badge, the 'paiza.io' logo, and a '新規コード' (New Code) button. A pink box highlights the '一覧' (Overview) button. Below the navigation bar, a message states 'Googleによる認証に成功しました。' (Authentication successful via Google). The main content area features tabs for 'おすすめ' (Recommended), '自分のコード' (My Code), 'お気に入り' (Favorites), '全て' (All), and 'Gists'. A row of language filters is displayed, with 'Python3' highlighted by a pink box. Below the filters, a search bar is visible. A pink box highlights the search results section, which shows '5 records found.' and a list of saved projects. The first project is titled 'テストプロジェクトです (Python3)' and is marked as private. The code editor for this project is open, displaying a Python script that calculates the average of numbers in a file.

```
1 # coding: utf-8
2 # Your code here!
3
4 data = open("File1", "r")
5
6 total = 0
7 num = 0
8
9 for line in data:
10     total = total + float( line )
11     num = num + 1
12     print(line)
13
14 data.close()
15
16 print(total)
17 print(num)
18
19 ave = total / num
20
21 print(ave)
22
```

一番上のメニューから「一覧」を選び、「Python3」を選択すると、保存されたプログラムが一覧されます。  
タイトルをクリックすると、編集画面に移動し、プログラミングを再開できます。

ちょっと本格的に  
Pythonプログラミングを  
やりたい



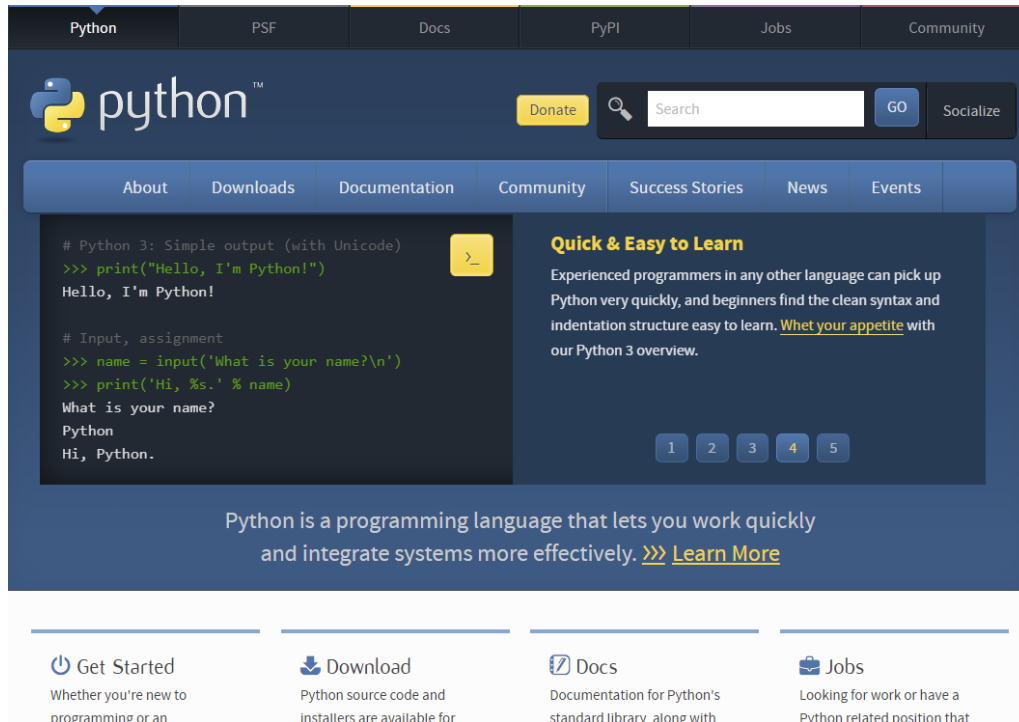
paiza.ioでは、プログラミングの基礎を体験することはできますが、ファイルを直接読み込むことや、計算した結果を「ファイル」で保存することができないなど、いくつか制限事項があります。

少し本格的にPythonを使ってみたい場合は、自分のPCにPythonをインストールして使います。

Pythonのインストール方法にもいくつかの選択肢がありますが、ここでは一番基本的なやり方を紹介します。

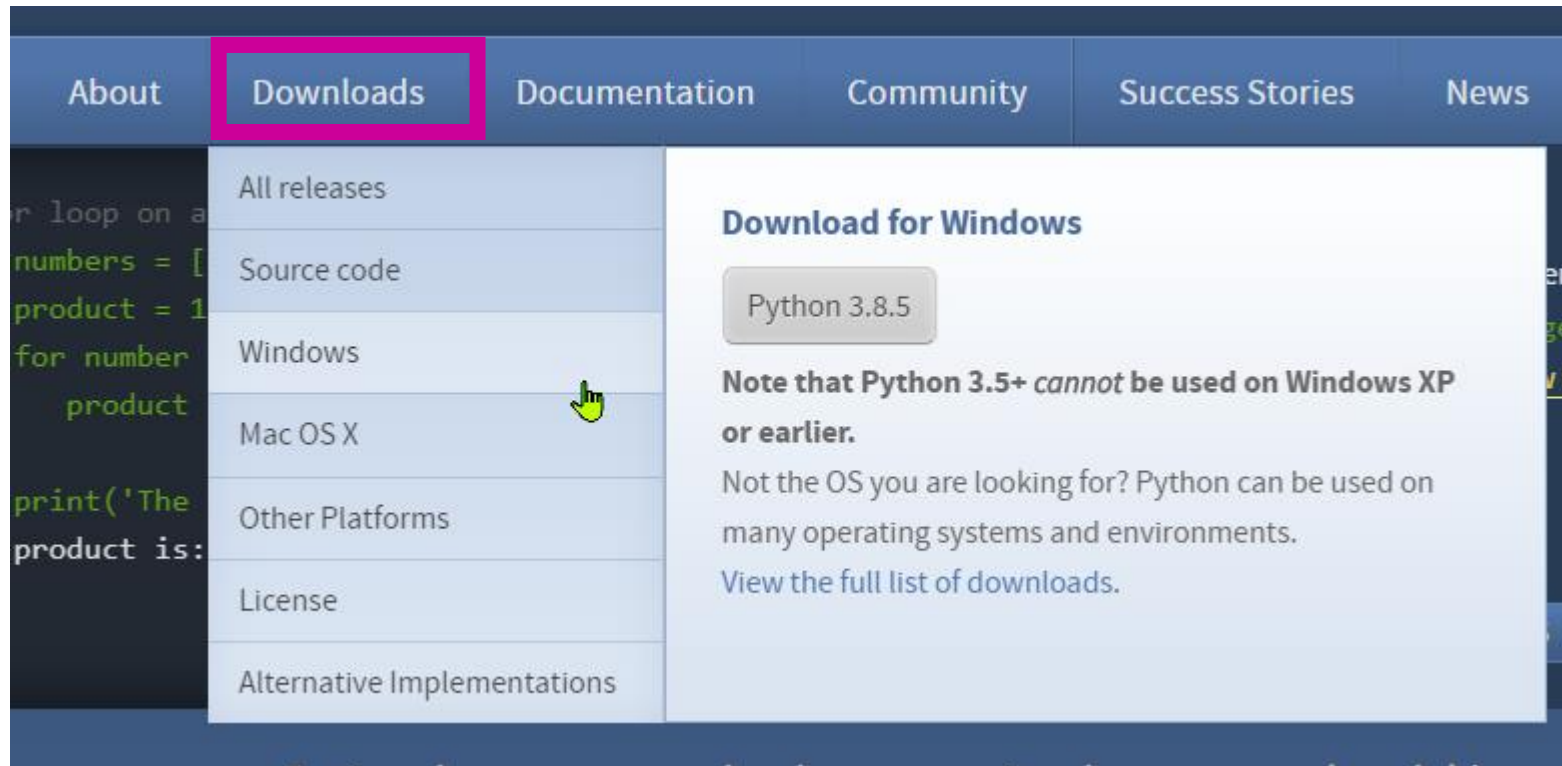
# Pythonのインストール

<https://www.python.org/>



Pythonのサイトに行きます。

# Pythonのインストール

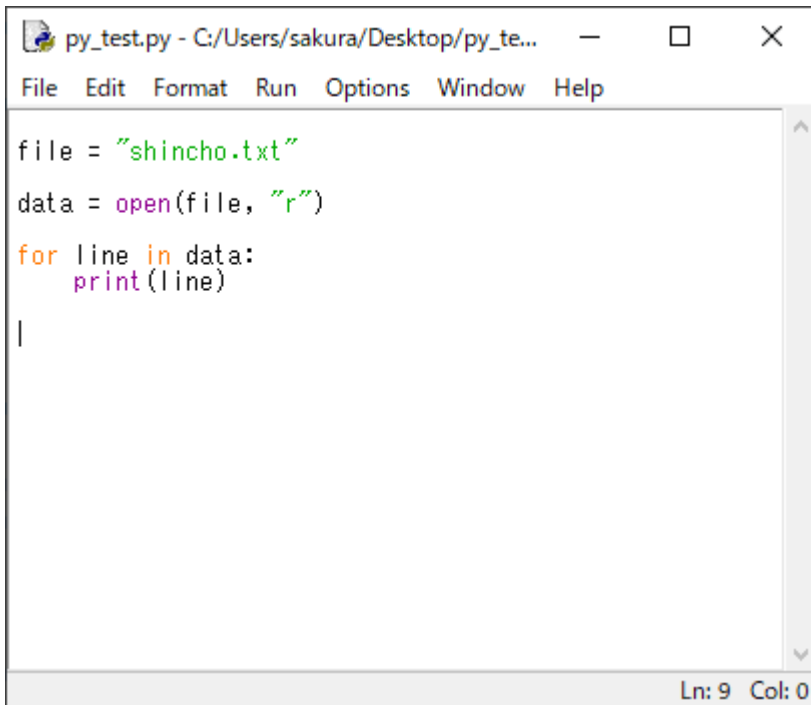


自分のPCのOSに合ったインストーラーをダウンロードします。

現在Pythonには、バージョン3.x系と2.x系がありますが、3.x系の最新版で特に問題ありません。

# 詳しいインストールの仕方、セットアップは、ネット上で情報を探してください。

WindowsやMac OSにインストールすると、IDLEと呼ばれる下記のようなコード編集ツールと一緒にインストールされます。



```
file = "shincho.txt"
data = open(file, "r")
for line in data:
    print(line)
|
```

見やすいコードの色分け表示をしてくれたりなど、効率的にプログラミングができます。

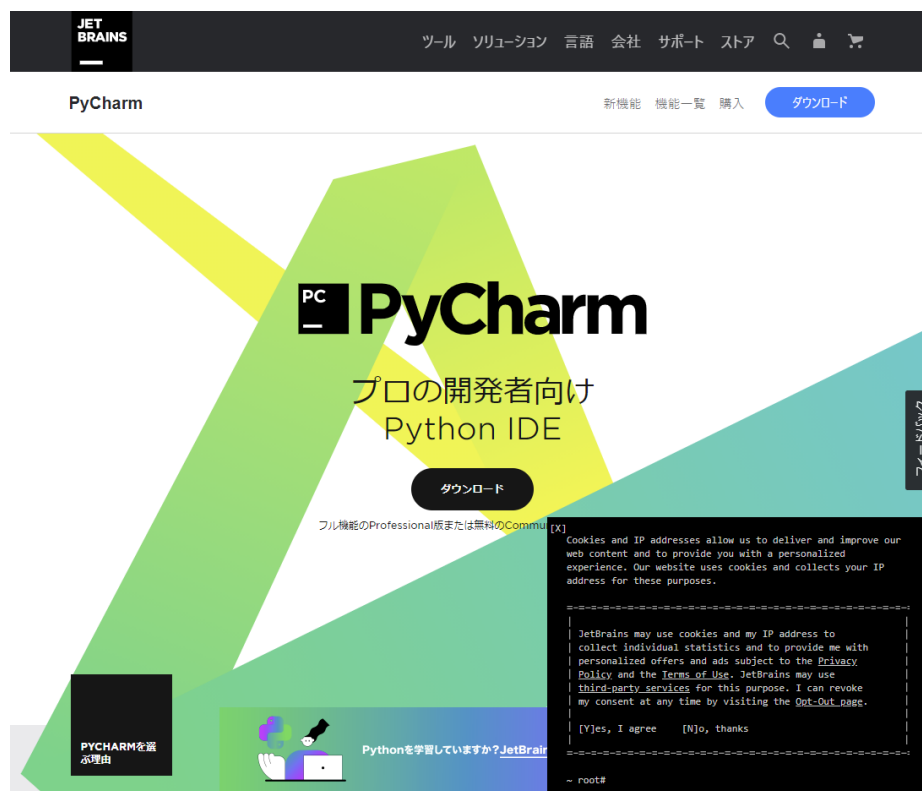
IDLE画面

さらに実践的に  
プログラミングがしたい

Pythonプログラミングを実践で使用するゆく場合は、高度なコード管理機能などがついた「開発環境」ソフトを使うと、効率よい開発が行えます。

おすすめのひとつが「PyCharm」です。

<https://www.jetbrains.com/ja-jp/pycharm/>



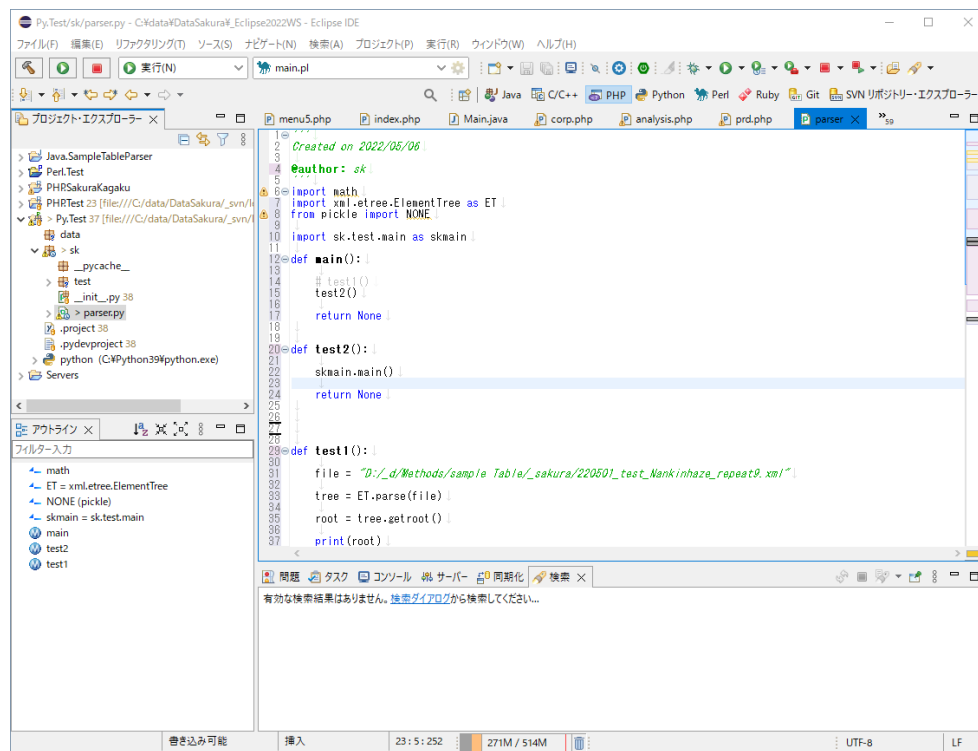
プロ向けの機能までたくさんついているため、いきなり使いこなすのは難しいですが、関係するコード間を行き来したり、変数名を一気に変えたり、エラーを事前に教えてくれたり、コードを補完してくれたりなど、実践的な開発には欠かせない機能が揃っています。

# 他のおすすめは「Eclipse」です



<https://mergedoc.osdn.jp/>

豊富なプラグインを組み合わせることで、Pythonだけでなく、Java、PHP、Rubyなど、様々な言語に対応できます。あらかじめ各種言語のプラグインを組み込み、日本語化されたパッケージ、Pleiades All in Oneが、上記リンクで配布されています。



コードの構造を表示する「アウトライン」という機能が、ほかのソフトに比べて使いやすく、個人的には一番好きです。

# 他のおすすめは「VSCode」です

<https://azure.microsoft.com/ja-jp/products/visual-studio-code/>



VSCode (Visual Studio Code)も、色々な言語に対応していて、人気の高いソフトです。様々なプラグインが活発に開発されているので、最新の技術トレンドをフォローしたい方に向いているかもしれません。

